



### مقدمه

هدف از این تمرین مهارت بیشتر شما با برنامه‌نویسی شیء‌گرای با استفاده از مفاهیم وراثت و چندریختی است. انتظار می‌رود از تکنیک‌های برنامه‌نویسی که تاکنون در کلاس درس فرا گرفته‌اید یا در هنگام تحویل حضوری تمرین‌ها به شما تذکر داده شده‌است به طور کامل در این تمرین استفاده کنید.

در این تمرین می‌خواهیم با استفاده از مهارت‌های برنامه‌نویسی که تاکنون کسب کرده‌اید برنامه‌ای بنویسیم که به خدای بازی مافیا درهم‌برهم برای مدیریت بازی کمک کند و تمام اتفاقات جریان بازی را ذخیره و برخی نتایج مراحل مختلف را اعلام کند.

### توضیح بازی مافیا

بازی مافیا یک بازی دسته جمعی متشکل از چندین نقش است. خدای بازی نقش‌ها را به بازیکنان اختصاص می‌دهد. بازیکنان چندین مرحله بازی را انجام می‌دهند تا زمانی که یکی از گروه‌های بازیکنان برنده بازی شود. نقش‌هایی که در این بازی باید پیاده‌سازی شوند عبارتند از:

joker, villager, detective, doctor, bulletproof, mafia, godfather, silencer

در بازی کلاسیک نقش‌ها به دو دسته مافیا<sup>۱</sup> و شهروند<sup>۲</sup> تقسیم می‌شوند. شهروندان از نقش بقیه بازیکنان خبر ندارند اما مافیایا یکدیگر را می‌شناسند. بازی در دو فاز شب و روز پیش می‌رود. در روز تمام افراد با یکدیگر بحث می‌کنند و سعی می‌کنند به‌درستی اشخاص مافیا را تشخیص بدهند، سپس رأی‌گیری اتفاق می‌افتد و فردی که بیشترین رأی را بیاورد از بازی حذف می‌شود. در فاز شب افراد مافیا رأی‌گیری می‌کنند و کسی که بیشترین رأی را بیاورد کشته می‌شود. در فاز شب برخی از نقش‌ها عملکرد متفاوتی دارند که در ادامه توضیح داده می‌شود. پس از اتمام فاز شب دوباره فاز روز شروع می‌شود و بازی ادامه می‌یابد. بازی در صورتی که تمام اشخاص مافیا کشته شوند به نفع شهروندان و اگر تعداد شهروندان کمتر یا مساوی تعداد اشخاص مافیا شود به نفع مافیا تمام می‌شود.

<sup>۱</sup> Mafia

<sup>۲</sup> Villager

## نقش‌های خاص شهروندان

**پزشک<sup>3</sup>:** خدای بازی هر شب پس از رأی‌گیری مافیا پزشک را بیدار می‌کند و از او می‌خواهد که یکی را نجات بدهد. اگر پزشک به‌درستی شخصی را که توسط مافیا کشته شده است انتخاب کند، او در آن شب کشته نمی‌شود.

**کارآگاه<sup>4</sup>:** خدای بازی هر شب کارآگاه را بیدار می‌کند و او می‌تواند شخصی را انتخاب کند از خدا بپرسد که آیا او مافیا است یا خیر. برنامه همواره جواب درست را به کارآگاه می‌دهد به جز در مواردی که شخص انتخاب شده **پدرخوانده<sup>5</sup>** باشد.

**ضد گلوله<sup>6</sup>:** ضد گلوله یک جان اضافه برای رأی‌گیری شب دارد؛ یعنی اگر یک شب توسط مافیا کشته شود نمی‌میرد و آن جان اضافه را از دست می‌دهد. اگر بار دیگری در شب توسط مافیا برای کشته شدن انتخاب شود، واقعاً می‌میرد. هیچ‌کس (حتی خود ضد گلوله) از این که قرار بوده توسط مافیا کشته شود مطلع نمی‌شود. اگر ضد گلوله در فاز روز برای کشته شدن انتخاب شود مانند دیگر شخصیت‌ها می‌میرد و جان اضافه‌اش به او کمکی نمی‌کند.

## نقش‌های خاص اعضای مافیا

**سایلنسر<sup>7</sup>:** سایلنسر هر شب یکی از اعضا را انتخاب می‌کند و این شخص در طول روز بعد نمی‌تواند حرف بزند یا در رأی‌گیری شرکت کند.

**پدرخوانده:** پدرخوانده از اعضای مافیاست و اگر در شب، کارآگاه عضویت او در مافیا را از برنامه بپرسد، با این که مافیا است، به کارآگاه اعلام می‌شود که او مافیا نیست.

## نقش‌های دیگر

**جوکر<sup>8</sup>:** جوکر در هیچ یک از گروه‌های مافیا و شهروندان قرار ندارد و همچنین قوانین برد و باخت بازی را به هم می‌ریزد. اگر جوکر در رأی‌گیری روز کشته شود، او به تنهایی برنده‌ی بازی می‌شود و تمام بازیکنان دیگر بازنده می‌شوند. اما اگر در شب توسط مافیا کشته شود اتفاقی نمی‌افتد و بازی ادامه پیدا می‌کند. فرض کنید **تنها یک جوکر** در بازی می‌تواند وجود داشته باشد.

## مافیای درهم درهم

مافیای درهم درهم یک نوع از بازی مافیا است که در پایان هر شب خدای بازی می‌تواند نقش‌ها را جابه‌جا کند! در ادامه جزئیات بازی و رفتار برنامه تعریف شده است.

---

<sup>3</sup> Doctor

<sup>4</sup> Detective

<sup>5</sup> Godfather

<sup>6</sup> Bulletproof

<sup>7</sup> Silencer

<sup>8</sup> Joker

## نحوه مدیریت استثناها<sup>9</sup>

در طول اجرای برنامه، ممکن است انواع خطا در برنامه شما رخ دهد. در این تمرین، انتظار می‌رود شما با پرتاب<sup>10</sup> کردن نمونه<sup>11</sup> هایی از کلاس‌هایی که ادامه ذکر خواهد شد و گرفتن<sup>12</sup> آن‌ها به خطاها رسیدگی کنید. این کلاس‌ها از کلاس `std::exception` ارث برده‌اند و شما باید تابع `what` این کلاس‌ها را `Override` کنید و در هنگام گرفتن این استثنا<sup>13</sup> ها، خروجی تابع `what` را در جریان خطای استاندارد<sup>14</sup> چاپ کنید. خطاهایی که انتظار می‌رود که شما در برنامه‌تان به آن‌ها رسیدگی کرده باشید، در شرح تمرین آمده‌اند.

## بخش نخست: ساخت بازی جدید

در ابتدا خدای بازی با دستور `create_game` یک بازی درست می‌کند. این دستور نام بازیکنانی که در بازی حضور خواهند داشت را به عنوان ورودی می‌گیرد.

### فرمت دستور:

```
create_game (<player_name>)*
```

### مثال:

```
create_game Rasta Hasan Hosna Bardia
```

## بخش دوم: اختصاص دادن نقش به بازیکن

خدا پس از ساختن بازی باید با استفاده از دستور `assign_role` نقش بازیکنان را به آن‌ها اختصاص دهد. این دستور نام بازیکن و نقش در نظر گرفته شده برای او را به عنوان آرگومان دریافت می‌کند.

### فرمت دستور:

```
assign_role <player_name> <role_name>
```

که در این دستور `<role_name>` یکی از کلمات زیر است:  
[joker, villager, detective, doctor, bulletproof, mafia, godfather, silencer]

### مثال:

```
assign_role Rasta mafia  
assign_role Hosna detective  
assign_role Bardia joker  
assign_role Hasan villager
```

---

<sup>9</sup> Exception Handling

<sup>10</sup> Throw

<sup>11</sup> Instance

<sup>12</sup> Catch

<sup>13</sup> Exception

<sup>14</sup> Standard Error Stream

نکات قابل توجه برای این دستور:

1. اگر قبل از اجرای این دستور، بازی توسط دستور `create_game` ایجاد نشده باشد، خطای زیر چاپ می شود:

`No game created`

2. اگر بازیکنی که نامش به برنامه داده شده است، در بازی وجود نداشت، خطای زیر چاپ می شود:

`User not joined`

3. اگر این دستور برای بازیکنی که قبلاً به او نقش اختصاص داده شده است صدا زده شود، نقش آن بازیکن، به نقش جدید داده شده، تغییر پیدا می کند.

## بخش سوم: شروع بازی

پس از اختصاص یافتن نقش ها به بازیکن ها، خدا می تواند با وارد کردن دستور زیر بازی را آغاز کند.

### فرمت دستور:

`start_game`

نکات قابل توجه برای این دستور:

1. اگر نقش تمام بازیکنان اختصاص نیافته باشد، خطای زیر چاپ می شود:

`One or more players do not have a role`

2. اگر این دستور پیش از ایجاد یک بازی (صدا زدن دستور `create_game`) فراخوانی شود خطای زیر چاپ می شود:

`No game created`

3. اگر این دستور وسط بازی فراخوانی شود خطای زیر چاپ می شود:

`Game has already started`

در صورتی که به تمام بازیکنان نقشی اختصاص داده شده باشد، ابتدا نقش بازیکنان مطابق خروجی زیر و به ترتیبی که نامشان در دستور `create_game` وارد شده است چاپ می شود:

Rasta: mafia  
Hasan: villager  
Hosna: detective  
Bardia: joker

سپس پیغام زیر چاپ شده و فرایند بازی شروع می شود.

Ready? Set! Go.

پس از شروع بازی، برنامه نقش هر بازیکن را می‌داند همچنین از کارهایی که در طول شب و در طول روز مجاز به انجام دادن آن‌ها هستند اطلاع دارد، بنابراین با ورود نام هر شخص به برنامه، وظیفه مربوط به آن بازیکن به طور خودکار انجام می‌پذیرد.

## بخش چهارم: فاز روز

### ● اعلام شروع روز

مطابق بازی مافیا، بازی در فازهای شب و روز ادامه پیدا می‌کند. در ابتدای هر روز، به منظور اعلام شروع روز پیغامی به فرمت زیر چاپ می‌شود. برای مثال پس از نمایش دادن پیام شروع بازی بلافاصله پیغام Day 1 چاپ می‌شود و برنامه منتظر ورودی‌های بعدی می‌شود.

### فرمت خروجی:

```
Day <day-num>
```

### مثال:

```
Day 1
```

### ● رأی‌گیری در روز

در ابتدای هر روز خدای بازی رأی‌گیری می‌کند. برای وارد کردن اطلاعات رأی‌ها در برنامه از فرمت زیر استفاده می‌شود.

### فرمت دستور:

```
<voter_name> <votee_name>
```

### مثال:

```
Hosna Bardia
```

آرگومان اول دستور، شخص رأی دهنده و آرگومان دوم دستور، شخصی که به او رأی داده شده است را مشخص می‌کند. به این ترتیب در مثال بالا یک رأی از طرف حسنا برای پردیا ثبت می‌شود. در صورتی که کسی بیش از یک بار رأی دهد، آخرین رایش به عنوان رأی نهایی ثبت می‌شود.

نکات قابل توجه برای این دستور:

1. هر دو نام voter\_name و votee\_name باید نام بازیکنان بازی باشند، در غیر این صورت خطای زیر چاپ می‌شود:

```
User not joined
```

2. اگر شخص رأی‌دهنده در شب گذشته توسط سایلنسر انتخاب شده باشد، خطای زیر چاپ می‌شود:

```
Voter is silenced
```

3. اگر شخصی که به او رأی داده شده است قبلاً کشته شده باشد خطای زیر چاپ می‌شود:

```
Votee already dead
```

## ● پایان رای گیری روز و اعلام نتایج

پس از اتمام رأی گیری خدای بازی دستور پایان رای گیری روز را وارد می کند. لزومی ندارد که تمامی بازیکنان رای داده باشند.

### فرمت دستور:

```
end_vote
```

پس از وارد شدن این دستور تعداد رأی های داده شده به هر شخص محاسبه و بازیکن با بیشترین تعداد رای کشته می شود. اگر تعداد رأی های چند بازیکن برابر باشند بازیکنی که نامش ترتیب الفبایی<sup>15</sup> کوچکتری دارد کشته می شود، برای مثال اگر تعداد رأی های Hosna، Rasta و Bardia برابر باشد، Bardia کشته می شود. پس از محاسبه بیشترین تعداد رأی، فرد کشته شده به صورت زیر چاپ می شود.

```
<user_name> died
```

پس از نمایش نام شخص کشته شده اگر بازیکن کشته شده در روز جوکر باشد، بازی پایان می یابد، جوکر برنده بازی می شود و پیام "Joker won" چاپ می شود. در غیر این صورت فاز شب شروع را اعلام می کند.

## بخش پنجم: فاز شب

### ● اعلام شروع شب

در ابتدای هر شب، پیامی به فرمت زیر چاپ می شود که نشانگر شروع شب و اتفاقات آن است.

```
Night <night-num>
```

با توجه به این که این بازی مافیا، درهم برهم است، دنبال کردن نقش افراد توسط خدای بازی سخت خواهد بود. برای همین برنامه در شروع شب لیست نام تمامی بازیکنان زنده ای که باید در طول شب بیدار شوند را با قالب زیر چاپ می کند.

### فرمت خروجی:

```
(<player_name>: <role>)*
```

### مثال:

```
Rasta: mafia  
Hosna: detective
```

<sup>15</sup> Lexicographical order:

در این ترتیب کد ASCII حروف مقایسه می شود. برای مثال ترتیب زیر را در نظر بگیرید:

$10 < 2 < A < a$

## ● اتفاقات در شب

تمام اتفاقات انجام شده در شب کارهایی هستند که یک بازیکن انجام می‌دهد و تاثیری بر بازیکن دیگر می‌گذارند.

در فاز شب اتفاقات زیر می‌افتد:

- رأی‌گیری مافیا
- نجات دادن پزشک
- پرس‌وجوی کارآگاه
- ساکت کردن ساینسر

در مافیا درهم‌برهم اتفاقات بالا به ترتیب صورت نمی‌گیرند. در ادامه خدای بازی هر بازیکن که باید در شب کاری انجام دهد را بیدار می‌کند و انتخاب او را در برنامه ثبت می‌کند. برای وارد کردن اتفاقات شب به برنامه کافی است خدای بازی نام بازیکنی که وظیفه‌ای در شب انجام می‌دهد و بازیکنی که وظیفه بازیکن اول بر او تاثیر می‌گذارد را به برنامه بدهد.

### فرمت دستور:

`<first_player> <second_player>`

به همین ترتیب هر نقش وظیفه خود را انجام می‌دهد. از میان افرادی که در شب وظیفه‌ای برای انجام دادن دارند تنها ساینسر است که هم در رأی‌گیری شبانه شرکت می‌کند و هم یکی از بازیکنان را ساکت می‌کند، بنابراین وقتی ساینسر بیدار می‌شود، در بار اول یکی از بازیکنان را ساکت می‌کند و از آن به بعد رای خود را به عنوان یکی از اعضای مافیا به یکی از بازیکنان می‌دهد. بنابر توضیحاتی که پیش‌تر آورده شد، تمام دستورات مربوط به وظایف بازیکنان از فرمت بالا پیروی خواهد کرد که اعمالی که first\_player انجام می‌دهد بر second\_player تاثیر می‌گذارد.

نکات قابل توجه برای این دستور:

1. اگر به عنوان آرگومان نخست نام شخصی آورده شود که در شب وظیفه‌ای ندارد (مثلا اعضای عادی شهروند) خطای زیر چاپ می‌شود.

User can not wake up during night

2. اگر آرگومان اول اسم کسی آورده شود که قبلا کشته شده است، خطای زیر چاپ می‌شود.

User is dead

در ادامه هر یک از اتفاقات شب دقیق‌تر توضیح داده شده است.

## ● رأی‌گیری مافیا

اگر بازیکن اول دستور مافیا (شامل پدرخوانده و ساینسر) باشد این دستور به معنی رای او به بازیکنی آرگومان دوم است. در طول شب تنها اعضای مافیا امکان رای دادن خواهند داشت. همچنین مانند روز بازیکنی که بیشترین رای را آورده است کشته خواهد شد و اگر رأی‌های مربوط به چند نفر از بازیکنان برابر بود بازیکنی که نامش **ترتیب الفبایی** کوچکتری دارد کشته می‌شود. توجه کنید که ساینسر از بار دوم رای خود را به عنوان یکی از اعضای مافیا به یکی از بازیکنان می‌دهد.

نکات قابل توجه برای این دستور:

1. هر یک از اعضای مافیا می‌تواند بیشتر از یک بار رأی بدهد اما تنها رأی آخر برای وی در نظر گرفته می‌شود.
2. هر دو آرگومان دستور باید نام بازیکنان بازی باشند، در غیر این صورت باید خطای زیر چاپ شود.

User not joined

3. اگر شخصی که به او رأی داده شده است قبلاً کشته شده باشد خطای زیر باید چاپ شود و بازیکن مافیا باید بار دیگر به یک بازیکن زنده رأی بدهد.

Votee already dead

## ● پرس و جوی کارآگاه

اگر بازیکن اول دستور کارآگاه باشد بازیکنی که نامش در آرگومان دوم آمده است را نجات می‌دهد. هر یک از کارآگاه‌های زنده باید یک بار بیدار شده و مافیا بودن یا نبودن یک بازیکن را بپرسند، بنابراین اگر بازیکن اول کارآگاه باشد، برنامه مافیا بودن یا نبودن بازیکن دوم را به او خواهد گفت. خروجی این دستور در صورتی که نقش بازیکن در آرگومان دوم از اعضای مافیا باشد، "Yes" و در غیر این صورت "No" است (اگر بازیکن در آرگومان دوم جوکر باشد هم خروجی "No" است). دقت داشته باشید که اگر بازیکن در آرگومان دوم نقش Godfather داشته باشد برنامه باید جواب "No" بدهد. نکات قابل توجه برای این دستور:

1. هر دو آرگومان دستور باید نام بازیکنان بازی باشند، در غیر این صورت باید خطای زیر چاپ شود.

User not joined

2. اگر یک کارآگاه بیشتر از یک بار پرس‌وجو انجام دهد خطای چاپ می‌شود.

Detective has already asked

3. اگر در آرگومان دوم نام شخصی که قبلاً کشته شده است بیاید، خطای زیر چاپ می‌شود و کارآگاه باید بار دیگر پرس‌وجو کند.

Suspect is dead

## ● نجات دادن پزشک

اگر بازیکن اول دستور پزشک باشد بازیکنی که نامش در آرگومان دوم آمده است را نجات می‌دهد. در فاز شب خدای بازی پزشک‌ها را بیدار می‌کند و امکان نجات دادن یک نفر را به هر یک از آن‌ها می‌دهد. اگر شخص نجات یافته توسط اعضای مافیا مورد سوءقصد قرار گرفته بود، نجات پیدا می‌کند و در آن شب نمی‌میرد. همچنین اگر هنوز رأی‌گیری شبانه اعضای مافیا هنوز اتمام پیدا نکرده باشد فردی که پزشک نجات می‌دهد در صورتی که پس از اتمام رأی‌گیری مورد سوء قصد قرار گرفت، در آن شب کشته نمی‌شود. این دستور باید برای هر پزشک زنده یک بار صدا زده شود بنابراین اگر بازیکن اول پزشک باشد، بازیکن دوم نجات داده می‌شود. اگر رأی‌گیری مافیا هنوز تمام نشده باشد، پس از تمام شدن در صورتی که پزشک قبلاً فردی را که به او سوءقصد شده است نجات داده باشد، آن بازیکن در آن شب کشته نمی‌شود.

نکات قابل توجه برای این دستور:

1. هر دو آرگومان دستور باید نام بازیکنان بازی باشند، در غیر این صورت باید خطای زیر چاپ شود.

User not joined

2. اگر پزشکی که یک بار بازیکنی را نجات داده است بار دیگر صدا زده شود خطای زیر چاپ می‌شود.



Doctor has already healed

3. اگر در آرگومان دوم نام شخصی که قبلاً کشته شده است بیاید، خطای زیر چاپ می شود و پزشک می تواند بار دیگر کسی را نجات دهد.

Patient is dead

4. چند پزشک می توانند یک نفر را برای نجات دادن انتخاب کنند. این حالت تفاوتی با حالتی که تنها یک پزشک آن شخص را نجات دهد ندارد.

## ● ساکت کردن سایلنسر

در فاز شب خدای بازی هر یک از سایلنسرها را بیدار می کند و امکان ساکت کردن یک نفر را به هر یک از آنان می دهد. اشخاص انتخاب شده توسط سایلنسرها در روز بعد نمی توانند در بحث ها شرکت کرده و نمی توانند رأی بدهند. سایلنسر باید پیش از رای دادن به یکی از بازیکنان حتماً یک نفر را ساکت کرده باشد. عمل ساکت کردن تنها در دفعه اول که سایلنسر در شب بیدار می شود صورت می پذیرد و در دفعات بعد مانند مافیا رای او به عنوان رای به کشته شدن یک نفر حساب می شود. این دستور باید برای هر سایلنسر زنده یک بار صدا زده شود.

نکات قابل توجه برای این دستور:

1. هر دو آرگومان ورودی باید نام بازیکنان بازی باشند، در غیر این صورت باید خطای زیر چاپ شود.

User not joined

2. در بار اول، اگر در آرگومان user\_name نام شخصی که قبلاً کشته شده است بیاید، خطای زیر چاپ می شود و سایلنسر می تواند بار دیگر کسی را ساکت کند.

Person is dead

3. از بار دوم به بعد، خطاها مطابق قسمت رای گیری مافیا اعمال می شود.

## ● پایان شب و اعلام اتفاقات

فاز شب با دستور end\_night با فرمت زیر پایان پیدا می کند:

end\_night

پس از آمدن دستور end\_night، ابتدا فردی که اعضای مافیا به جانش سوء قصد کردند به صورت زیر چاپ می شود.

Mafia tried to kill <user\_name>

سپس شب پایان یافته و بازی وارد فاز روز می شود. از روز دوم به بعد علاوه بر شماره روز، گزارش اتفاقات شب گذشته مانند فرمت زیر چاپ می شود:

Day <day\_num>  
<user\_name> was killed  
Silenced (<user\_name>)\*

اگر شخصی که توسط مافیا مورد سوء قصد قرار گرفت حداقل توسط یکی از پزشک‌های بازی نجات یافته باشد به عنوان فرد کشته‌شده چاپ نمی‌شود و پیام اول (Killed) به طور کلی چاپ نمی‌شود. اگر تمام سایلنسرها در طی بازی کشته شده باشند و کسی در شب گذشته ساکت نشده باشد پیام دوم (Silenced) چاپ نمی‌شود. اگر شخصی توسط چند سایلنسر ساکت شده باشد تنها یک بار در لیست افراد ساکت شده چاپ می‌شود. همچنین این لیست باید بر اساس حروف الفبا مرتب شده باشد و نام بازیکن‌ها با اسپیس از یکدیگر جدا شده باشند. دقت کنید اگر کسی که مورد سوء قصد مافیا قرار گرفته است ضدگلوله بوده و هیچ پزشکی او را نجات نداده باشد و فرد برای اولین بار در شب گذشته مورد سوء قصد قرار گرفته باشد، جان اضافه‌اش از بین می‌شود و در روز به عنوان فرد کشته شده نمایش داده نمی‌شود.

## بخش ششم: درهم برهم سازی

در مافیای درهم برهم خدای بازی می‌تواند یک بار در هر شب پس از اتمام تمام اتفاقات مربوط به فاز شب، نقش ۲ نفر از بازیکنان را با هم عوض می‌کند. این کار با دستور زیر صورت می‌گیرد:

```
swap_character <user_name1> <user_name2>
```

به این ترتیب نقش دو بازیکن با نام‌های user\_name1 و user\_name2 با یکدیگر تعویض می‌شوند و تمام اطلاعات شخصیت‌های بازیکنان نیز تعویض می‌شود مثلاً اگر ضد گلوله جان اضافه خود را از دست داده باشد پس از تعویض شخصیت نیز فردی که شخصیت ضد گلوله را می‌پذیرد جان اضافه نخواهد داشت. علاوه بر این تعویض شخصیت تنها برای بازیکن‌های زنده قابل انجام است. خدا بلافاصله بعد پایان هر شب حداکثر یک بار می‌تواند این دستور را استفاده کند.

نکات قابل توجه برای این دستور:

1. هر دو نام user\_name1 و user\_name2 باید نام بازیکنان بازی باشند، در غیر این صورت باید خطای زیر چاپ شود.

User not joined

2. اگر هر یک از بازیکنان user\_name1 یا user\_name2 قبلاً کشته شده باشند، خطای زیر چاپ می‌شود و خدا می‌تواند بار دیگر برای تعویض شخصیت دو بازیکن اقدام کند. اگر فرد در شب فعلی کشته شده باشد نیز شامل این خطا می‌شود.

User is dead

3. اگر خدا در یک شب بیش از یک بار از این دستور استفاده کند پیغام خطای زیر چاپ می‌شود.

Characters already swapped

4. اگر این دستور پیش از اتمام شب صورت بگیرد، خطای زیر چاپ می‌شود.

Can't swap before end of night

5. اگر این دستور هنگام رأی‌گیری روز وارد شود خطای زیر چاپ می‌شود.

Voting in progress

## بخش هفتم: وضعیت بازی

خدای بازی در هر زمان می‌تواند با وارد کردن دستور `get_game_state` تعداد اعضای مافیا و شهروندان را به دست آورد.

### فرمت دستور:

```
get_game_state
```

### فرمت خروجی:

```
Mafia = <number_of_mafia>
Villager = <number_of_villager>
```

جوکر در تعداد شهروندان محاسبه نمی‌شود.

## بخش هشتم: پایان بازی

بازی تا زمانی ادامه می‌یابد که یکی از دو تیم مافیا یا شهروندان یا جوکر برنده بازی شود. همان‌طور که پیش‌تر هم گفته شد شروط بردن هر یک از تیم‌ها به صورت زیر است:

- اگر تمام اعضای مافیا کشته شوند شهروندان برنده بازی می‌شوند و پیغام زیر چاپ می‌شود.  
`Villagers won`
- اگر تعداد اعضای شهروندان کمتر یا مساوی تعداد اعضای مافیا شود، مافیا برنده بازی می‌شود و پیغام زیر چاپ می‌شود.  
`Mafia won`
- اگر جوکر در رأی‌گیری روز کشته شود، جوکر برنده بازی می‌شود و پیغام زیر چاپ می‌شود.  
`Joker won`

## نکات تکمیلی

- در پایان رشته‌هایی که در توابع `what` در استثناها بر می‌گردانید حتماً کاراکتر `'\n'` وجود داشته باشد.
- مدیریت خطاهایی که در شرح تمرین توضیح داده‌شد کافیت و خطاهای دیگری که ممکن است رخ دهد، در هنگام تحویل سنجیده نمی‌شود.
- طراحی درست وراثت، رعایت سبک برنامه‌نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد. برای مثال استفاده از `if` یا `switch case` برای تشخیص نوع زیرکلاس یک کلاس پدر نشان‌دهنده طراحی نادرست وراثت است.
- در طول فاز روز ممکن است همه بازیکنان رای ندهند اما در طول فاز شب تضمین می‌شود همه بازیکنان زنده شب حداقل یک‌بار رای می‌دهند (سایلنسر حداقل دوبار رای می‌دهد).

## نحوه‌ی تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی **zip** با نام `A6-<SID>.zip` در صفحه‌ی elearn درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ است، نام پرونده‌ی شما باید `A6-810198999.zip` باشد.
- **دقت کنید** که پرونده‌ی zip آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما (از جمله Makefile) باشد و از zip کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید.
- برنامه شما باید حتماً طراحی شیء‌گرا داشته باشد و حتماً در آن از وراثت و چندریختی برای ساده کردن پیاده‌سازی و حذف حالت‌بندی‌ها استفاده شده باشد. این موضوع قسمت بزرگی از نمره‌ی شما را شامل می‌شود.
- **دقت کنید** که پروژه‌ی شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد C++11 استفاده می‌کنید.
- **دقت کنید** که نام پرونده‌ی اجرایی شما باید `mafia.out` باشد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.